****

**A**

**PROJECT REPORT**

**ON**

**“BAKERY MANAGEMENT SYSTEM”**

**SUBMITTED BY: Tanuja Dhanraj Gaikwad (2124UCEF1040)**

**SUBJECT:** **C++ PROGRAMMING**

**Under the guidance of: Miss. Ishwari Tirse**

**Department of Computer Science and Engineering**

**Sanjivani Rural Education Society**

**SANJIVANI UNIVERSITY**

**KOPARGAON - 423603, DIST: AHEMDNAGAR**

**2024-2025**

**Department of Computer Science and Engineering**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SR**  **NO.** | **CONTEXT** | **PAGE NO.** |
| **1.** | **INTRODUCTION** | **3** |
| **2.** | **CODE** | **8** |
| **3.** | **OUTPUT** | **9** |
| **4.** | **CONCLUSION** | **10** |

**INTRODUCTION**

Bakery Management System

The Bakery Management System class is the core of the application, containing functions for various operations. By encapsulating the functionalities within a class, we achieve better organization and make the code more modular and reusable.

1. Inventory Tracking: The system enables users to add and view bakery items with essential details like name, price, and stock quantity. This feature ensures that staff can maintain a comprehensive and up-to-date inventory.

2. Order Processing: The application allows for quick and easy order placement, automatically updating stock levels and calculating costs, thereby simplifying the checkout process for customers.

3. Stock Management: Users can update the quantity of items as needed, making it easy to replenish stock and ensure popular items are always available.

**CODE**

#include <iostream>

#include <vector>

#include <string>

using namespace std;

// Struct to hold bakery item details

struct Item {

int id;

string name;

double price;

int quantity;

};

// Class for Bakery Management System

class BakeryManagementSystem {

private:

vector<Item> inventory;

public:

// Function to add an item to inventory

void addItem(int id, string name, double price, int quantity) {

Item newItem = { id, name, price, quantity };

inventory.push\_back(newItem);

cout << "Item added successfully!" << endl;

}

// Function to display all items in inventory

void displayItems() {

cout << "\n--- Bakery Inventory ---\n";

for (const auto& item : inventory) {

cout << "ID: " << item.id

<< ", Name: " << item.name

<< ", Price: $" << item.price

<< ", Quantity: " << item.quantity << endl;

}

}

// Function to update stock of an item

void updateStock(int id, int quantity) {

for (auto& item : inventory) {

if (item.id == id) {

item.quantity += quantity;

cout << "Stock updated successfully!" << endl;

return;

}

}

cout << "Item not found!" << endl;

}

// Function to place an order

void placeOrder(int id, int orderQuantity) {

for (auto& item : inventory) {

if (item.id == id) {

if (item.quantity >= orderQuantity) {

item.quantity -= orderQuantity;

cout << "Order placed! Total cost: $" << (orderQuantity \* item.price) << endl;

} else {

cout << "Insufficient stock!" << endl;

}

return;

}

}

cout << "Item not found!" << endl;

}

};

// Main function

int main() {

BakeryManagementSystem bakery;

int choice, id, quantity;

string name;

double price;

while (true) {

cout << "\n--- Bakery Management System ---\n";

cout << "1. Add Item\n";

cout << "2. Display Items\n";

cout << "3. Update Stock\n";

cout << "4. Place Order\n";

cout << "5. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

cout << "Enter item ID: ";

cin >> id;

cout << "Enter item name: ";

cin >> name;

cout << "Enter item price: ";

cin >> price;

cout << "Enter item quantity: ";

cin >> quantity;

bakery.addItem(id, name, price, quantity);

break;

case 2:

bakery.displayItems();

break;

case 3:

cout << "Enter item ID to update: ";

cin >> id;

cout << "Enter quantity to add: ";

cin >> quantity;

bakery.updateStock(id, quantity);

break;

case 4:

cout << "Enter item ID to order: ";

cin >> id;

cout << "Enter quantity to order: ";

cin >> quantity;

bakery.placeOrder(id, quantity);

break;

case 5:

cout << "Exiting system.\n";

return 0;

default:

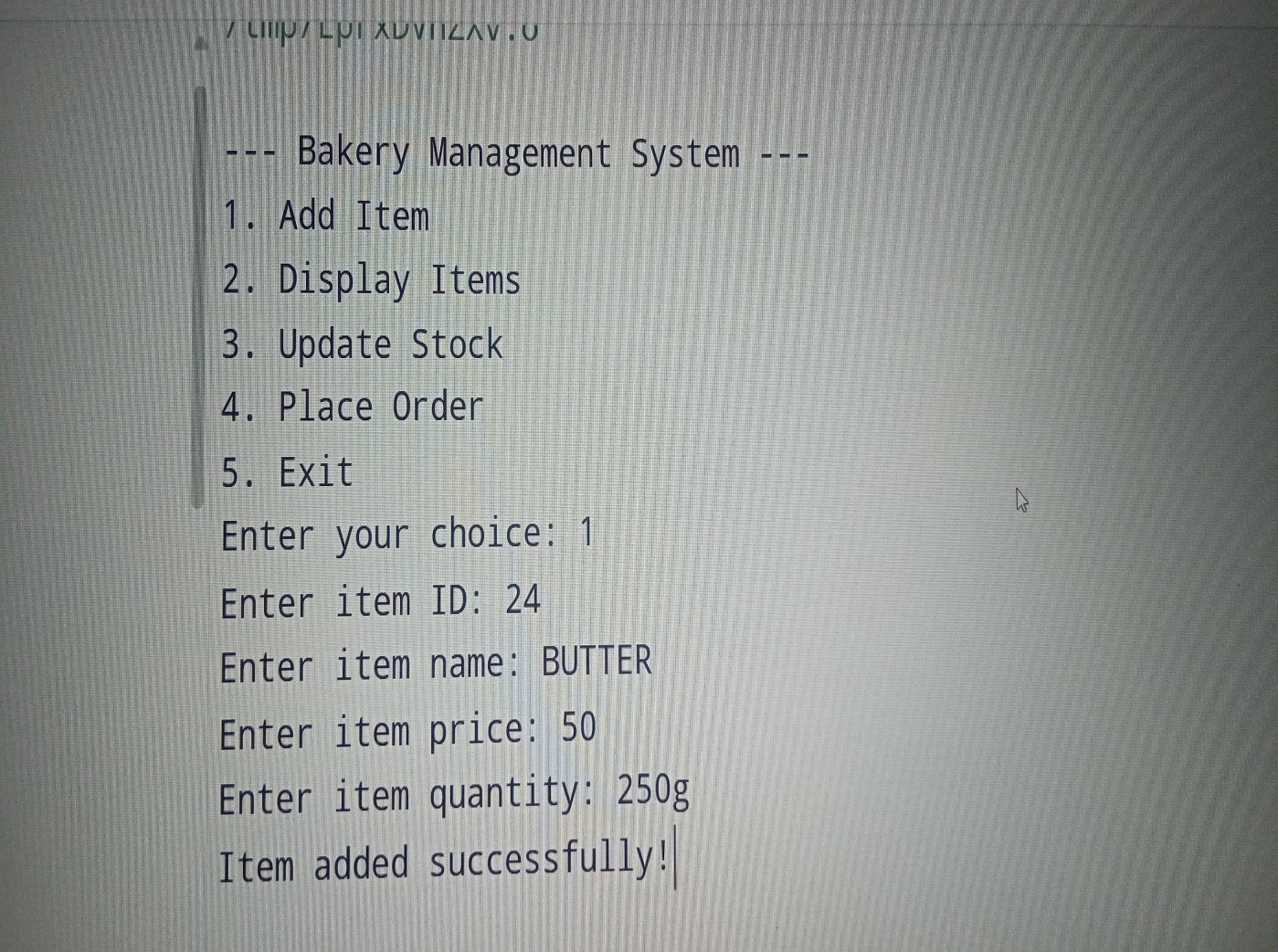
cout << "Invalid choice. Try again.\n";

}

}

}

**OUTPUT**

****

**CONCLUSION**

In the Bakery Management System in C++ demonstrates a simple yet effective approach to managing a bakery's inventory and order processing needs through a console-based application. By structuring the system into organized components such as the Item structure and the Bakery Management System class, the code showcases basic principles of object-oriented programming, including encapsulation and modularity.

The system’s main features—adding items, displaying inventory, updating stock, and processing orders—are all core functions that any bakery would require to manage its operations efficiently. Through a user-friendly menu-driven interface, the application provides an intuitive way for bakery staff to interact with the system, making it adaptable for real-world scenarios.